Non-negative Tensor Factorization with missing data for the modeling of gene expressions in the Human Brain

Nielsen, Søren Føns Vind; Mørup, Morten

Published in:
Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2014)

Link to article, DOI:
10.1109/MLSP.2014.6958919

Publication date:
2014

Document Version
Peer reviewed version

Non-Negative Tensor Factorization with Missing Data for the Modeling of Gene Expressions in the Human Brain

Søren Føns Vind Nielsen and Morten Mørup*

Abstract

Non-negative Tensor Factorization (NTF) has become a prominent tool for analyzing high dimensional multi-way structured data. In this paper we set out to analyze gene expression across brain regions in multiple subjects based on data from the Allen Human Brain Atlas [1] with more than 40 % data missing in our problem. Our analysis is based on the non-negativity constrained Canonical Polyadic (CP) decomposition where we handle the missing data using marginalization considering three prominent alternating least squares procedures; multiplicative updates, column-wise, and row-wise updating of the component matrices. We examine three gene expression prediction scenarios based on data missing at random, whole genes missing and whole areas missing within a subject. We find that the column-wise updating approach also known as HALS performs the most efficient when fitting the model. We further observe that the non-negativity constrained CP model is able to predict gene expressions better than predicting by the subject average when data is missing at random. When whole genes and whole areas are missing it is in general better to predict by subject averages. However, we find that when whole genes are missing from all subjects the model based predictions are useful. When analyzing the structure of the components derived for one of the best predicting model orders the components identified in general constitute localized regions of the brain. Non-negative tensor factorization based on marginalization thus forms a promising framework for imputing missing values and characterizing gene expression in the human brain. However, care also has to be taken in particular when predicting the genetic expression levels at a whole region of the brain missing as our analysis indicates that this requires a substantial amount of subjects with data for this region in order for the model predictions to be reliable.

Keywords: Non-negative Tensor Factorization, CandeComp/PARAFAC, CP, Non-negative Matrix Factorization, Missing Values, Marginalization.

*This work was supported by the Lundbeck Foundation, grant no. R105-9813.

1 Introduction

Tensor or multi-way array decompositions have become popular tools for analyzing and summarizing multi-way data in part due to the recent increased storage and computation capabilities[2]. A tensor is the generalization of a vector or matrix to multi-dimensional objects, i.e. a vector is a 1st order tensor and a matrix is a 2nd order tensor. The two most widely used tensor decompositions include the Tucker model and the Canonical Polyadic(CP) model also denoted the PARAFAC/CandeComp models. The Tucker model decomposes an Nth order tensor into N matrices called loadings, and a core-array, which is also an order Nth tensor that defines how the loadings of each mode interact. If we restrict the core-array of the Tucker model to be diagonal with value one in the diagonal elements, we arrive at the CP model that in contrast to the Tucker model in general has a unique minimizer of the cost function (up to scale and permutation of the components)[3]. Furthermore, if we restrict the loadings (and core-array) to be non-negative the problem becomes a Non-Negative Tensor Factorization (NTF), see also [4]. Tensor factorization has seen use in fields including computational chemistry, neuroimaging, signal processing, and web mining[2, 5, 6]. When solving for the NTF problem the alternating least squares procedure is commonly used where the tensor decomposition problem is recast into multiple standard non-negative matrix factorizations (NMF) problems. We note that NMF and thus NTF is celebrated due to its part based representation[7] and relation to clustering [8].

A common problem when decomposing data sets including tensors is handling missing data[9, 10, 11, 12, 13]. Even matrix factorization with missing entries is non-trivial and has been shown to be NP-Hard [14]. In general for decompositions there are two approaches to handling missing data: 1) *Imputation* in which the missing values are estimated iteratively which is related to the Expectation Maximization algorithm. This type of approach is often the easiest to implement and requires almost no changes in the model. 2) *Marginalization* or *weighted regression*, in which the missing values are ignored during the optimization of the cost function. The latter approach has been shown to handle more data being missing including systematic patterns of missing data [9]. For the case of unconstrained tensor factorization marginalization has been shown to scale well to larger datasets where recovery of the underlying components of synthetic data was possible when up to 99% of the data is missing [11]. Using an alternating least squares estimation strategy for NTF based on the CP model the N-way toolbox available at: `http://www.models.life.ku.dk/nwaytoolbox` [15] for MATLAB is equipped with an imputation procedure whereas marginalization for NTF in the context of Bayesian inference has been considered in [16]. All parameters at once optimization approaches for NTF with missing data has been considered in [12].

In this paper we implement and compare three prominent alternating least squares NTF estimation approaches for missing data based on marginalization; *Multiplicative Update*, *Hierarchical Alternating Least Squares* and *Active Set*. We apply the best performing of these methods for the analysis of the Allen Human Brain Atlas [1] which is a large data set with 43% missing values containing measurements of genetic expressions across brain regions in multiple subjects. In particular, we investigate the ability of the non-negative CP model for missing data based on marginalization to predict genetic

expressions in the brain in three scenarios; subjects missing data at random, subjects missing measurements of a particular gene across the entire brain, and subjects missing measurements in a particular region of the brain. We contrast the predictions of the non-negative CP model to a simple procedure where missing entries are predicted as the average over subjects having the considered entries present.

2 Methods

The CP model is for a 3rd order tensor, $\mathcal{X}^{I\times J\times K}$ defined by

$$\mathcal{X}^{I\times J\times K} \approx \sum_d \mathbf{a}_d^I \circ \mathbf{b}_d^J \circ \mathbf{c}_d^K, \quad x_{ijk} \approx \sum_d a_{id} b_{jd} c_{kd}, \tag{1}$$

in which $\mathbf{a}_d^I$ is the d'th column of the loading $\mathbf{A}^{I\times D}$, D is the number of components in the model and $\circ$ denotes the outer product, i.e., $(\mathbf{u}^I \circ \mathbf{w}^J)_{ij} = u_i w_j$. Notice here that the choice of D influences the model. The model can equivalently be written in matrix notation as

$$\mathbf{X}_{(1)} \approx \mathbf{A}(\mathbf{C}\odot\mathbf{B})^\top,\ \mathbf{X}_{(2)} \approx \mathbf{B}(\mathbf{C}\odot\mathbf{A})^\top,\ \mathbf{X}_{(3)} \approx \mathbf{C}(\mathbf{B}\odot\mathbf{A})^\top,$$

in which $\mathbf{X}_{(1)}^{I\times JK}$ is the data tensor $\mathcal{X}^{I\times J\times K}$ matricized on the first mode, $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ are the mode specific loadings from the CP-model, and $\odot$ is the Kathri-Rao product (column-wise Kronecker product), such that $\mathbf{A}^{I\times D} \odot \mathbf{B}^{J\times D} = \mathbf{S}^{IJ\times D}$, with $s_{j+J(i-1),d} = a_{id} b_{jd}$. For further details on tensor nomenclature see also [5, 2]. One of the most common approaches to find the loading matrices is by alternatingly solving the above matrix factorization problems for each mode. If we use the squared Frobenius norm as objective function the first subproblem becomes

$$\min_{\mathbf{A}} ||\mathbf{X}_{(1)} - \mathbf{A}(\mathbf{C}\odot\mathbf{B})^\top||_F^2, \quad \text{s.t.} \quad a_{id} \geq 0, \quad \forall i, d \tag{2}$$

This can be viewed as a standard NMF problem in which we are interested in estimating the component matrix $\mathbf{A}$.

In this paper we consider three prominent ways to solve the above NMF problem; *Multiplicative Updates* (MU), *Hierarchical Alternating Least Squares* (HALS) and *Active Set*. A recent review of these NMF approaches can be found in [17]. In the following we will describe the algorithms in detail and their update rules. We will use the following standard notation:

$$E = \frac{1}{2}||\mathbf{X} - \mathbf{WH}||_F^2 = \frac{1}{2}\sum_{i,j}\left(x_{ij} - \sum_d w_{id} h_{dj}\right)^2, \tag{3}$$

in which $\mathbf{W}$ is the factor we want to determine and $\mathbf{H}$ is the fixed factor in the alternating procedure. Furthermore, if we introduce missing entries using marginalization the cost function is given by

$$E = \frac{1}{2}\sum_{i,j} r_{ij}\left(x_{ij} - \sum_d w_{id} h_{dj}\right)^2, \tag{4}$$

in which r_{ij} is an indicator variable, which is 1 if the entry is present and 0 if the entry is missing.

2.1 Multiplicative Update

The multiplicative updates (MU) was proposed in [7, 18]. MU is based on gradient descent with a specific choice of step size. Consider the gradient of the cost function in (3) with respect to an element $\mathbf{W}_{id}$ which is given by

$$\nabla w_{id} = (\mathbf{WHH}^\top)_{id} - (\mathbf{XH}^\top)_{id}. \tag{5}$$

To ensure that the cost function is decreasing we want to take a step in the negative direction of the gradient, and by setting the step size, $\eta_{id} = \frac{\mathbf{W}_{id}}{(\mathbf{WHH}^\top)_{id}}$, we arrive at the multiplicative update rule

$$\mathbf{W}_{id} \leftarrow \mathbf{W}_{id} \frac{(\mathbf{XH}^\top)_{id}}{(\mathbf{WHH}^\top)_{id}}. \tag{6}$$

We note that the above MU must include fixes for the denominator becoming zero as well as the update being stuck at $\mathbf{W}_{id} = 0$, see also [19] and references therein.

Using this procedure for the marginalization approach given in (4) for handling missing data gives the following updates[20]

$$\mathbf{W}_{id} \leftarrow \mathbf{W}_{id} \frac{((\mathbf{R} * \mathbf{X})\mathbf{H}^\top)_{id}}{\sum_j (\mathbf{WH})_{ij} \mathbf{R}_{ij} \mathbf{H}_{dj})}, \tag{7}$$

in which $*$ denotes element-wise multiplication, and $\mathbf{R}$ is the present data indicator matrix. NMF-algorithms have been shown to improve convergence speed by updating the same factor multiple times before updating the next, see also [21]. We note that $\mathbf{W}$ only enters once in the fraction in the denominator (7). Therefore by pre-computing $(\mathbf{R} * \mathbf{X})\mathbf{H}$ with cost $O(IJD)$ and all inner products between components in $\mathbf{H}$ weighted by the indicator matrix $\mathbf{R}$ with cost $O(IJD(D+1))$, we can with little extra computation effort update $\mathbf{W}$ several times as the calculation of the denominator using the pre-computations has cost $O(ID^2)$ and the ratio of numerator to denominator multiplied to the elements of $\mathbf{W}$ has cost $O(ID)$. To select the number of times $\mathbf{W}$ should be updated (repetitions) we use a balancing heuristic, i.e. the computational complexity of all the pre-computations should equal the complexity of the update step times the number of repetitions. Writing the computational complexities in terms of I, J, D and the number of repetitions, Q, gives

$$O(IJD + IJD(D+1)) = O(Q(ID^2 + ID)) \Rightarrow Q \approx J.$$

2.2 Hierarchical Alternating Least Squares

The hierarchical alternating least squares (HALS) approach was first described briefly in [22] and further considered in [23]. The HALS exploits that the cost function can be

rewritten component-wise as

$$E = \frac{1}{2} \sum_{i,j} \left((x_{ij} - \sum_{d \neq \ell} (w_{id} h_{dj})) - w_{i\ell} h_{\ell j} \right)^2 .$$

Taking the derivative with respect to the column element $w_{i\ell}$ yields

$$\frac{\partial E}{\partial w_{i\ell}} = \sum_j w_{i\ell} h_{\ell j}^2 + \sum_j h_{\ell j} \left(\sum_{d \neq \ell} w_{id} h_{dj} \right) - \sum_j x_{ij} h_{\ell j},$$

and setting this to zero thereby solving for $w_{i\ell}$ gives

$$w_{i\ell} = \frac{\sum_j x_{ij} h_{\ell j} - \sum_j h_{\ell j} (\sum_{d \neq \ell} w_{id} h_{dj})}{\sum_j h_{\ell j}^2}.$$

The non-negativity constraint is implemented by truncating negative values to zero, i.e.,

$$w_{i\ell} \leftarrow \max \left(0, \frac{\sum_j x_{ij} h_{\ell j} - \sum_j h_{\ell j} (\sum_{d \neq \ell} w_{id} h_{dj})}{\sum_j h_{\ell j}^2} \right). \tag{8}$$

Considering the marginalization for HALS we again take the derivative of (4) with respect to the specific column-element $w_{i\ell}$ which gives

$$\frac{\partial E}{\partial w_{i\ell}} = \sum_j r_{ij} w_{i\ell} h_{\ell j}^2 + \sum_j r_{ij} h_{\ell j} \left(\sum_{d \neq \ell} w_{id} h_{dj} \right) - \sum_j r_{ij} x_{ij} h_{\ell j}.$$

This leads to the following update rule for $w_{i\ell}$

$$w_{i\ell} \leftarrow \max \left(0, \frac{\sum_j r_{ij} x_{ij} h_{\ell j} - \sum_j r_{ij} h_{\ell j} (\sum_{d \neq \ell} w_{id} h_{dj})}{\sum_j r_{ij} h_{\ell j}^2} \right). \tag{9}$$

For the above HALS type update, the most costly computational operation is determining the second term in the numerator, which is also the only term that includes elements from $\mathbf{W}$. It can be interpreted as the reconstruction $\mathbf{WH}$ without the ℓ'th component times the ℓ'th component from $\mathbf{H}$ weighted with the indicator matrix, $\mathbf{R}$. Rearranging the sums in the second term gives

$$w_{i\ell} \leftarrow \max \left(0, \frac{\sum_j r_{ij} x_{ij} h_{\ell j} - \sum_{d \neq \ell} w_{id} \sum_j r_{ij} h_{\ell j} h_{dj}}{\sum_j r_{ij} h_{\ell j}^2} \right) \tag{10}$$

Like MU, the above HALS marginalization based updates can also be speed up by updating $\mathbf{W}$ several times pre-computing $\sum_j r_{ij} x_{ij} h_{\ell j}$, $\sum_j r_{ij} h_{\ell j} h_{dj}$, and $\sum_j r_{ij} h_{\ell j}^2$. Considering the balancing heuristic from before we again obtain $Q \approx J$.

2.3 Active Set

The Active Set (AS) algorithm is based on finding the set of variables, that if found by unconstrained optimization, violate the non-negativity constraint. If the true active set is known the solution to the problem is simply solved by unconstrained optimization (eg. by the normal-equations) where the variables in the active set are ignored and set to zero. The AS algorithm was adapted for NMF in [24] in which it was called Fast Non-Negativity-Constrained Least Squares (FNNLS). In this version the NMF problem of identifying $\mathbf{W}$ is solved one row of $\mathbf{W}$ at a time. The algorithm iteratively finds the most important variable based on the gradient, includes this in the model and backtracks variables that no longer should be part of the solution. When alternatingly solving for loadings of each mode this method can be speed up by saving the active set from previous iterations of the alternating least squares procedure [24]. Missing values are handled by only considering a row of $\mathbf{R}$ at a time and including only columns j of $\mathbf{X}$ and $\mathbf{H}$ in which the elements $\mathbf{R}_{ij} = 1$ in the subproblem (3) [9].

3 Results and Discussion

We benchmark the three algorithms MU, HALS, and AS with and without inner iterations when solving for the NTF with missing data based on marginalization. Subsequently, we analyze the Allen Human Brain Atlas data set and consider predicting genetic expression levels under different missing conditions.

3.1 A comparison of MU, HALS, and AS

We consider Multiplicative Updates (MU), Hierarchial Alternating Least Squares (HALS), a combination-approach where we alternated between MU and HALS steps and finally the Active Set (AS) algorithm. For the first three subproblem solvers we tested two different kind of configurations, one in which the number of inner iterations was 1 and another where the number of inner iterations was set according to the balancing heuristic. NTF as for NMF is in general influenced by initialization ([17]) so we ran all models multiple times with different initializations scaled to reflect the magnitude of the data.

We tested the methods on a synthetic data set, $\mathcal{X}^{1000\times 50\times 25}$, which was generated from a true 5 component CP-model with and without noise. To give a realistic picture of the performance of the algorithms we treated 40 % of the data as missing at random. All methods were restarted with 5 different initializations followed by one MU update for each mode. The result can be seen in figure 1 and figure 2. Algorithms that are subscripted 1 here run with a single inner iteration pr. mode, and subscripted J run with the number of inner iterations given by the balancing heuristic.

As we can see, AS, HALS$_J$ and MU+HALS $_J$ converge almost equally in terms of number of iterations, but the two algorithms with HALS require less CPU-time. MU converges slowly both when considering progress in terms of CPU-time and iterations. Furthermore, we see that taking multiple inner iterations as expected improves convergence compared to the single inner iteration configuration. As a result, the following

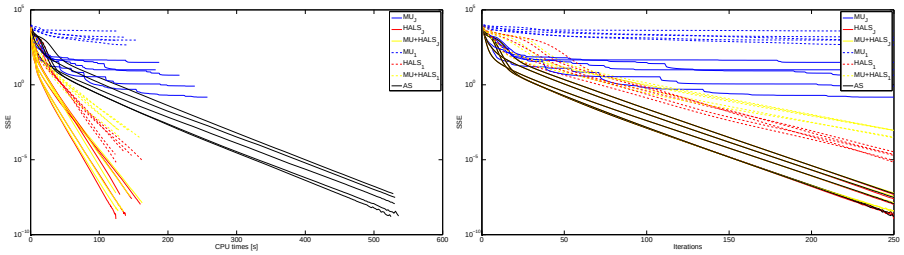

Figure 1: Sum of squared error (SSE) as a function of CPU time and number of iterations. All algorithms were forced to run 250 iterations. No noise was added to the data

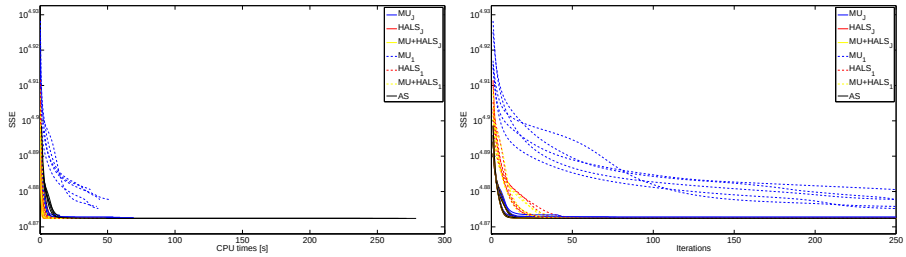

Figure 2: Sum of squared error (SSE) as a function of CPU time and number of iterations. All algorithms were forced to run 250 iterations. Gaussian noise ($\sigma^2 = 0.1$) was added to the data

analysis will be based on the HALS including inner iterations. Furthermore, we have included two stopping criteria for the algorithm - the change in cost function is smaller than 10^{-6} times the value of the cost function, and a maximum number of outer iterations to be 250.

On the synthetic data with noise we conducted a validation procedure to see if our CP model could capture the true number of components in the synthetic data set. We used the root mean squared error (RMSE) as a measure of the predictive performance, and for each $D = \{1, 2, ..., 10\}$ we restarted the algorithm 5 times. The results of the validation can be seen in Table 1. We can see that the model is able to capture the underlying number of components in the synthetic data.

D	Training Error	Test Error
1	0.2745 $\pm$ 8e-13	0.3541 $\pm$ 8e-10
2	0.2627 $\pm$ 2e-8	0.3396 $\pm$ 3e-8
3	0.2544 $\pm$ 7e-8	0.3293 $\pm$ 6e-7
4	0.2489 $\pm$ 4e-8	0.3228 $\pm$ 4e-7
5	0.2442 $\pm$ 3e-8	**0.3172 $\pm$ 4e-7**
6	0.2440 $\pm$ 1e-6	0.3175 $\pm$ 4e-6
7	0.2438 $\pm$ 3e-6	0.3177 $\pm$ 2e-5
8	0.2435 $\pm$ 4e-6	0.3180 $\pm$ 6e-6
9	0.2433 $\pm$ 3e-6	0.3183 $\pm$ 8e-6
10	0.2431 $\pm$ 4e-6	0.3186 $\pm$ 1e-5

Table 1: Above the RMSE as a function of number of components, D, in the CP-model is shown. For each D the model was restarted 5 times, and values in the table display the mean $\pm$ the standard deviation of the mean

3.2 Predicting genetic expressions in the human brain

The cells in our body all have different functions and 'tasks'. These functions are dictated by which genes in the DNA that are *expressed*, and these expression levels can be measured by micro-arrays. The Allen Human Brain Atlas is a collection of gene expression data from six different brains [1] (available from [25]). This provides a unique gene-fingerprint for each sampled region taken from a subjects brain. In the data set, labels have been applied to each sample defining which 'major' area of the brain the sample is taken from. We preprocessed the data by taking the mean expression values over all the samples from the same area within a subjects brain. Notice though that data is not present from all areas in the six subjects. We formatted the data into a tensor, $\mathcal{X}^{N_{genes} \times N_{areas} \times N_{subjects}}$, for further analysis. Here $N_{genes} \approx 58000, N_{areas} = 414$ and $N_{subjects} = 6$. Notice that after our preprocessing approximately 43% of the entries in the tensor were missing.

We presently set out to investigate how well genetic expressions can be predicted in this data by the non-negativity constrained CP model using marginalization. We consider three prediction scenarios; 1) predicting data missing at random, 2) predicting whole genes completely missing from a subject, 3) predicting areas completely missing from a subject. These three scenarios constitute prediction challenges of increased complexity. We used a hold-out set to test the predictive performance of the model for different values of $D = \{1, 5, 10, 15, 20, 25, 30\}$. We chose the hold-out set in the following three different ways. In the first approach we removed 10 percent of the present entries at random. In the second approach we created a 6-fold cross-validation partitioning of all genes, where each fold defined a subjects respective held out genes. In the third approach we took out entire areas, such that each area, which was in more than one subject, was in the test set exactly once. This meant that we were certain that all areas were present in at least one subject in the training set.

The results of the analysis is given in Table 2. We compare our model to predicting

	$D = 1$	$D = 5$	$D = 10$	$D = 15$	$D = 20$	$D = 25$	$D = 30$
Random entries							
Training	0.703 ± 0	0.520 ± 5.9e-06	0.455 ± 2.8e-05	0.418 ± 6.4e-04	0.393 ± 3.3e-04	0.376 ± 3.5e-04	0.360 ± 3.2e-04
Test (*T1-5*)	0.777 ± 0	**0.575 ± 7.9e-06**	**0.507** ± 1.4e-05	**0.468 ± 6.2e-04**	**0.442 ± 4.3e-04**	**0.425 ± 4.3e-04**	**0.409 ± 4.5-04**
Mean (*T1-5*)	**0.678 ± 0**	0.680 ± 1.3e-05	0.680 ± 2.4e-05	0.678 ± 5.9e-05	0.678 ± 7.8e-05	0.678 ± 6.4e-05	0.680 ± 1.2e-04
Test (*T6*)	**0.765 ± 0**	**0.577 ± 4.6e-06**	**0.496 ± 2.6e-04**	**0.473 ± 4.3e-04**	**0.445 ± 9.8e-04**	**0.430 ±3.3e-04**	**0.420 ± 8.3e-04**
Mean (*T6*)	0.790 ± 0	0.790 ± 7.1e-04	0.790 ± 5.5e-05	0.789 ± 3.3e-04	0.789 ± 3.2e-04	0.788 ± 4.8e-04	0.789 ± 4.4e-04
Genes							
Training	0.705± 7.4e-05	0.521 ± 6.8e-04	0.455 ± 6.3e-04	0.417 ± 4.3e-04	0.392 ± 1.9-04	0.375 ± 3.1-04	0.361 ± 3.1e-04
Test (*T1-5*)	0.802 ± 4.6e-04	0.732 ± 5.5e-02	0.721 ± 3.0e-02	**0.670 ±2.1e-02**	0.693 ± 3.6e-03	0.777 ±4.970e-02	0.771 ± 3.438e-02
Mean (*T1-5*)	**0.670 ± 2.3e-04**	**0.670 ± 2.6e-04**	**0.669 ± 2.216e-04**	**0.670 ± 5.9e-04**	**0.671 ± 5.3e-04**	**0.671 ± 4.5e-04**	**0.670 ± 3.8e-04**
Test (*T6*)	**0.804 ± 5.6e-04**	0.834 ± 9.7e-02	**0.804 ± 4.9e-02**	**0.705 ± 2.8e-02**	**0.716 ± 1.3e-02**	**0.821 ± 5.2e-02**	**0.786 ± 1.4e-02**
Mean (*T6*)	0.830 ± 6.9e-04	**0.829 ± 1.6e-03**	0.827 ± 8.0e-04	0.828 ± 1.5e-03	0.831 ± 1.2e-03	0.829 ± 1.1e-03	0.828 ± 8.3e-04
Areas							
Training	0.668 ± 2.6e-04	0.494 ± 6.3e-04	0.429 ± 8.62e-04	0.392 ± 3.63e-04	0.368 ± 5.9e-04	0.352 ± 5.5e-04	0.338 8.4e-04
Test (*T1-5*)	0.785 ± 8.5e-04	5.68e05 ± 5.7e05	7.21e04 ± 7.2e04	2.12e02 ± 1.3e02	4.50e02 ± 2.2e02	1.16e11 ± 1.2e11	1.8e07 ± 1.8e07
Mean (*T1-5*)	**0.716 ± 1.2e-03**	**0.718 ±1.3e-03**	**0.716 ± 1.1e-03**	**0.717 ± 2.2e-03**	**0.717 ± 1.5e-03**	**0.715 ± 9.9e-04**	**0.716 ± 1.3e-03**
Test (*T1*)	0.735 ± 6.2e-03	0.596 ± 1.8e-02	**0.557 ± 2.2e-02**	0.615 ± 2.8e-02	0.788 ± 5.6e-02	1.02 ± 1.7e-01	1.29 ± 2.3e-01
Mean (*T1*)	**0.580 ± 1.5e-03**	**0.587 ± 2.0e-03**	0.582 ± 3.9e-03	**0.584 ± 4.0e-03**	**0.586 ± 5.8e-03**	**0.579 ± 2.5e-03**	**0.578 ± 3.6e-03**
Test (*T2*)	0.807 ± 3.5e-03	3.11 ± 2.5	2.87 ± 2.2	0.758 ± 3.4e-02	2.02 ± 6.2e-01	1.93 ± 4.9e-01	2.31 ± 5.5e-01
Test (*T3*)	0.876 ± 1.9e-03	1.621 ± 9.6e-01	1.43 ± 6.7e-01	2.5 ± 9.3e-01	4.27 ± 1.7	4.53 ± 8.9e-01	1.01e+01 ± 3.7
Test (*T4*)	0.938 ± 7.1e-03	1.59 ± 7.2e-01	6.82 ± 4.0	4.89 ± 2.2	1.58e+01 ±4.0	3.60e+01 ± 1.8e+01	7.75e+07 ± 7.7e+07
Test (*T5*)	0.765 ± 3.0e-03	8.57e+05 ± 8.6e+05	1.088e+05 ± 1.1e+05	3.21e+02 ± 1.9e+02	6.80e+02 ± 3.36e+02	1.8e+11 ± 1.8e+11	2.49e+04 ± 2.5e+04

Table 2: Above the RMSE error as a function of number of components, D, in the CP-model is shown. For each D the model was restarted 5 times, and values in the table display the mean ± the standard deviation of the mean. For the case of missing areas, we have subdivided the table into the categories T1,T2,T3,T4 and T5 that indicate the test error for the areas that were missing in 1,2,3,4 and 5 subjects respectively. For the case of missing genes and missing entries at random we list two different errors, T1-5 and T6

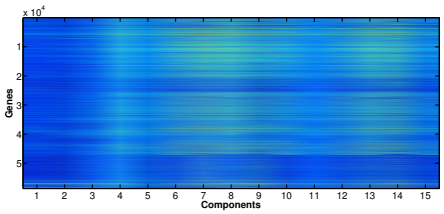

Figure 3: The final results for the gene-mode. The 15 components are visualized in a heat map according to the gene expression levels in a $\log(1 + x)$-scale. Values have been normalized to be between zero (blue) and one (red)

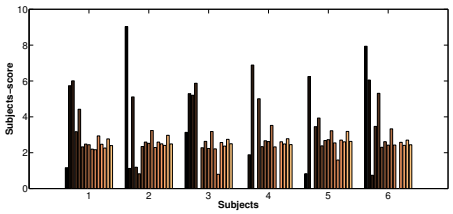

Figure 4: The final results for the subject-mode visualized in a bar plot. The 15 components are each assigned a color, and the height of the bar indicates the components "importance" in the given subject.

by the mean over other subjects with the entry present. In the case of random missing entries and missing genes it occurs that a gene in an area is missing in all subjects after holding out the test set. For these entries we predict by the mean over both areas and subjects. In the table the error $T1, .., T6$ denotes the error on the entries that are missing in $1, .., 6$ subjects respectively, such that $T1 - T5$ denotes predictions made where data in at least one subject is present whereas $T6$ denotes the predictions made for entries where all subjects have the data missing. From the results it can be seen that when data is missing at random the non-negative CP model performs best at predicting and significantly better than predicting using subject averages (except for $D = 1$) while the predictions improve as the model becomes more expressive (i.e., the best predictions are attained for $D = 30$.). The model does not predict hold-out genes better than predicting by the subject average in the entries present in at least one subject, though we see a superior performance by the model in entries that are missing in all subjects. When predicting areas the model is only better than predicting by the subject average when data in the analysis of a region is present in five other subjects when using a 10 component model.

As 15 components performed well in all scenarios of prediction we ran the non-negative CP model on the whole dataset with $D = 15$, with 5 restarts and we chose the solution with lowest SSE (0.460 with standard deviation $9.7e - 4$) . In Figure 3, 4 and 5, we see each of the 3 modes and their respective estimated components. The values of the gene and area mode has been scaled to be between 0 and 1 whereas the subject mode has been upscaled accordingly. From the analysis it can be seen that there is variability in the components across subjects while most of the identified components concentrate on specific brain regions.

4 Conclusion

In this paper we addressed the problem of missing data in the non-negative constrained CP model using marginalization. We compared three prominent alternating least squares approaches for solving the NMF subproblem, namely Multiplicative Update, Hierarchial Alternating Least Squares (HALS) and the Active Set procedure. We found that the HALS was the computationally best performing method when using multiple inner iterations. When analyzing a dataset of genetic expressions for multiple subjects in the

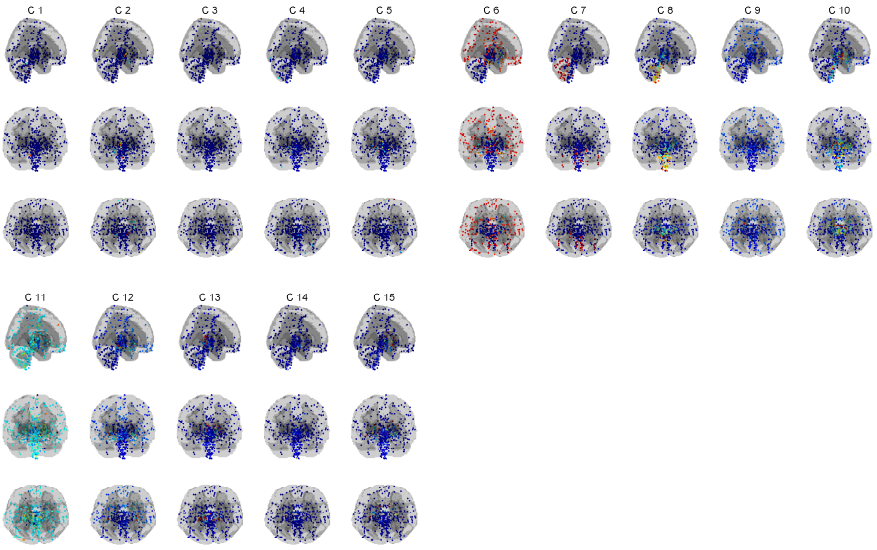

Figure 5: Brain-area mode from the final model is visualized from three angles in MNI coordinates. Values have been scaled to be between 0 (blue) and 1 (red).

human brain we found that the non-negative constrained CP improved on prediction when compared to predicting by the subject average when data was missing at random. When whole genes were missing predictions only improved where data was missing in all subjects whereas the predictions for whole regions missing were unreliable unless five other subjects included data for the region. The non-negative CP model identified components reflecting localized genetic profiles and their subject specific variation. Future work should investigate the reliability of the identified components as well as investigate their relation to brain structure and function.

References

[1] Michael J. Hawrylycz et al. An anatomically comprehensive atlas of the adult human brain transcriptome. *NATURE*, 489(7416):391–399, 2012.

[2] Morten Morup. Applications of tensor (multiway array) factorizations and decompositions in data mining. *WILEY INTERDISCIPLINARY REVIEWS-DATA MINING AND KNOWLEDGE DISCOVERY*, 1(1):24–40, 2011.

[3] Joseph B Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138, 1977.

[4] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[5] TG Kolda, BW Bader, and JP Kenny. Higher-order web link analysis using multilinear algebra. *FIFTH IEEE INTERNATIONAL CONFERENCE ON DATA MINING, PROCEEDINGS*, pages 242–249, 2005.

[6] Evrim Acar and Bülent Yener. Unsupervised multiway data analysis: A literature survey. *Knowledge and Data Engineering, IEEE Transactions on*, 21(1):6–20, 2009.

[7] Daniel D. Lee and H. Sebastina Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401(6755):789, 1999.

[8] Chris HQ Ding, Xiaofeng He, and Horst D Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *SDM*, volume 5, pages 606–610. SIAM, 2005.

[9] G. Tomasi and R. Bro. Parafac and missing values. *CHEMOMETRICS AND INTELLIGENT LABORATORY SYSTEMS*, 75(2):163–180, 2005.

[10] Georgio . *Practical and computational aspects in chemometric data analysis*. PhD thesis, Department of Food Science The Royal Veterinary and Agricultural University, Denmark, 2006.

[11] Evrim Acar, Morten Mørup, Tamara G. Kolda, and Daniel M. Dunlavy. Scalable tensor factorizations with missing data. *Proceedings of the 10th SIAM International Conference on Data Mining, SDM 2010*, pages 701–712, 2010.

[12] Jean-Philip Royer, Nadege Thirion-Moreau, Pierre Comon, et al. Nonnegative 3-way tensor factorzation taking into account possible missing data. *EUSIPCO-2012*, pages 1–5, 2012.

[13] Kohei Hayashi, Takashi Takenouchi, Tomohiro Shibata, Yuki Kamiya, Daishi Kato, Kazuo Kunieda, Keiji Yamada, and Kazushi Ikeda. Exponential family tensor factorization for missing-values prediction and anomaly detection. In *Data*

Mining (ICDM), 2010 IEEE 10th International Conference on, pages 216–225. IEEE, 2010.

[14] François Glineur and Nicolas Gillis. Low-rank matrix approximation with weights or missing data is np-hard. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1149–1165, 2011.

[15] C. A. Andersson and R. Bro. The n-way toolbox for matlab. *Chemometrics and Intelligent Laboratory Systems*, 52(1):1–4, 2000.

[16] Mikkel Schmidt and Shakir Mohamed. Probabilistic non-negative tensor factorisation using markov chain monte carlo. *EUSIPCO-2009*, 2009.

[17] N. Gillis. The Why and How of Nonnegative Matrix Factorization. *ArXiv e-prints*, January 2014.

[18] DD Lee and HS Seung. Algorithms for non-negative matrix factorization. *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 13*, 13:556–562, 2001.

[19] Eric C. Chi and Tamara G. Kolda. On tensors, sparsity, and nonnegative factorizations. *SIAM JOURNAL ON MATRIX ANALYSIS AND APPLICATIONS*, 33(4):1272–1299, 2012.

[20] Yun Mao and Lawrence K. Saul. Modeling distances in large-scale networks by matrix factorization. *Proceedings of the 2004 ACM SIGCOMM Internet Measurement Conference, IMC 2004*, pages 278–287, 2004.

[21] Nicolas Gillis and Francois Glineur. Accelerated multiplicative updates and hierarchical als algorithms for nonnegative matrix factorization. *NEURAL COMPUTATION*, 24(4):1085–1105, 2012.

[22] Rasmus Bro. *Multi-way Analysis in the Food Industry*. PhD thesis, Chemometrics Group, Food Technology Department of Dairy and Food Science, Royal Veterinary and Agricultural University, 1998.

[23] Andrzej Cichocki and Anh-Huy Phan. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE TRANSACTIONS ON FUNDAMENTALS OF ELECTRONICS COMMUNICATIONS AND COMPUTER SCIENCES*, E92A(3):708–721, 2009.

[24] R. Bro and S. DeJong. A fast non-negativity-constrained least squares algorithm. *JOURNAL OF CHEMOMETRICS*, 11(5):393–401, 1997.

[25] Allen Institue for Brain Science. Allen Human Brain Atlas, 2014.